A Data Mining Grid Scheduling Model Based on Colored-hierarchy Petri Net

Zhang Hui

Department of Physics, Guangdong University of Education

zhangtougao2010@163.com

***Abstract*-Dynamics and granularity in data mining grid scheduling (DMGS) are greatly influenced by resources. DMGS is defined as a workflow, combined with colored-hierarchy Petri Nets (CHPN) structure of the scheduling variable scheduling algorithm, which contains the scheduling problem into job scheduling layer, sub-job scheduling layer, task scheduling layer and sub-task scheduling layer. Within the four layers, job scheduling will be decomposed from top to bottom. Each layer is based on Petri net state transfer. The results show that HCPN can be effective in DMGS.**

Keywords-Colored-hierarchy; Petri Net; Data Mining Grid; Scheduling

I. INTRODUCTION

Data mining grid scheduling (DMGS) is a dynamic scheduling process with the granularity of jobs may fluctuate due to resources [1]. Therefore, they need to be redistributed. A job usually includes several related operations or tasks. DMGS is a complex process [2]. A clear and intuitive scheduling model is necessary to define the process, analyse and evaluate their performance.

DMGS defines a job as a workflow and is able to integrate with other algorithms. Thus, it has been widely studied and applied [3]. Scheduling model based on Petri nets can be easily adjusted according to the resource allocation and real-time to support the dynamic merge sub-tasks, split, update, and modification [4].

Petri Net is one of several mathematical modelling languages for the description of distributed systems, which is a directed bipartite graph [5-8]. A Petri Net includes of places, transitions and directed arcs. Different elements have the specific meanings [9]. For example, the nodes represent transitions and places, the directed arcs imply which places are pre- and /or post conditions for which transitions.

Arcs in Petri Net run from a place to a transition or vice versa, never between places or between transitions [10]. Two types of places are here. One is input places that an arc runs to a transition from the places. The other is output places that arcs run from a transition to the places. Places often contain a natural number of tokens, of which are over the places [11, 12].

Execution of Petri Net is nondeterministic [5, 13, 14]. That means multiple transitions are enabled at the same time, which would be fire. If a transition is enabled, it might be fire. Multiple tokens may be present in the net at anywhere and anytime [12]. Petri nets are well suited for modelling the concurrent activities of distributed systems such as planning and scheduling system, process management system, warehouse distributing system, etc [5, 6, 10, 12, 13, 15].

This paper presents a DMGS algorithm which is based on colored-hierarchy Petri net (CHPN) to monitor the job assignment and scheduling process accurately as well as to evaluate and analyse the performance. This algorithm decomposes the scheduling processes from top to bottom. The top level describes the process of the scheduling, while the low level focuses on the scheduling granularity. The purpose of the two-hierarchy scheduling approach is to reduce the scheduling time and algorithm complexity [16, 17, 18, 19]. This algorithm possesses some advantages such as high coupling, good performance and effective through using the CHPN approach

II. COLORED-HIERARCHY PETRI NETS (CHPN)

CHPN scheduling defines the tasks planning and scheduling network through top to down decomposition [20, 21]. Its purpose is to reduce the complexity and adaptive to change its structure according to the granularity so as to support the dynamic scheduling [10]. There are some definitions as follows:

Definition 1: CHPN is defined as a set with eight elements:

$$Grid = \{P, F, D, C, I, O, K, M\},$$

where,

$$P = \{p, T\}$$

p keeps the jobs with different statuses and granularities [12].

$$T = \{t_s \cup t_c\}$$

is a set of statuses. t_s represents the status transfer. t_c means the complex status transfer. F is the set of arc which determines the input and output [22].

D describes the set of different colors that to differentiate the jobs such as jobs, sub-jobs, tasks and sub-tasks [23].

C denotes the color function.

$$C : P \cup T \to \phi(D)$$

$\phi(D)$ is the coefficient set of colors. I, O denote the functions of status transfer input and output arc respectively.

$$(p', t') \in (P \times T)$$

have to meet

$$I(p', t') \in [C(p')_{ms} \to C(t')_{ms}]_L$$

and

$$I(p',t')=0$$

is based on

$$(p',t')\notin F.$$

K denotes the capacity that determines the number of jobs parallel processing in the scheduling network [24]. M means the initiative status.

$$M: p \rightarrow D_{ms}.$$

Definition 2: $Grid$ will transferred from one status to another one is based on :

$$\forall p' \in^{\otimes} t : M(p') \geq O(p',t') \wedge \forall p' \in t^{\otimes} : M(p') + I(p',t') \leq K(p')$$

p'_{start} , p'_{end} means the start and end status.

$$^{\otimes}p'_k = \varnothing \wedge p_l^{\otimes} = \varnothing \wedge p'_k \neq p'_l$$

Definition 3: Any complex status in the $Grid$ could be extended as a sub-network:

$$S\_Grid_i = \{P_i, F_i, D_i, C_i, I_i, O_i, K_i, M_i\}$$

where,

$$P_i = \{p_i, T_i\}$$

$$\{t_i^k \mid 1 \leq k \leq n\}$$

meets

$$U_{k=1}^{n} I(p_{start}^{i}, t_i^k) = U_{k=1}^{u} I(p_k, t_i)$$

$$\{p_k \mid 1 \leq k \leq u\} = ^{\otimes} t_i.$$

III. DMGS BASED ON CHPN

DMGS algorithm based on CHPN can be divided into four layers [25, 26, 27]. They are job scheduling layer, sub-job scheduling layer, task scheduling layer and sub-task scheduling layer. They will be detailed demonstrated in the following sections.

A. Job Scheduling layer

This layer mainly executes the execution of jobs. The jobs statuses are running, waiting, completed and failed.

Let:

$$Grid = \{P, F, D, C, I, O, K, M\}$$

where,

$$P = \{p_i \mid 1 \leq i \leq 5\}, T = \{t_i \mid 1 \leq i \leq 5\}$$

$$D = \{d_i \mid 1 \leq i \leq 9\}$$

d_i means:

TABLE I EXPLANATION OF d_i

d_i	Explanations
d_1	Jobs submitted from the users. $C(p_1)=\{d_1\}$, $C(t_1)=\{d_1\}$
d_2	Jobs waiting for processing. $C(p_2)=\{d_2\}$, $C(t_2)=\{d_2,d_9\}$
d_3	Jobs are under processing. $C(p_3)=\{d_3\}$, $C(t_3)=\{d_3\}$
d_4	Jobs are waiting for check. $C(p_4)=\{d_4\}$, $C(t_4)=\{d_4\}$
d_5	Jobs are correct. $C(p_5)=\{d_5,d_6,d_7\}$, $C(t_5)=\{d_5\}$
d_6	Jobs are completed. $C(p_5)=\{d_5,d_6,d_7\}$, $C(t_6)=\{d_6\}$
d_7	Jobs are failed. $C(p_5)=\{d_5,d_6,d_7\}$, $C(t_7)=\{d_7\}$
d_8	Jobs are returned to the users. $C(p_7)=\{d_8\}$
d_9	Jobs TOKEN with limited numbers that can be run parallel. $C(p_6)=\{d_9\}$

The jobs transfer in this layer is as figure 1.

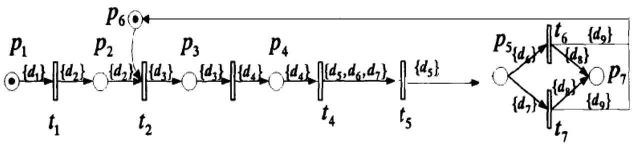


Fig. 1. Jobs Status Transfer in Job Scheduling Layer

B. Sub-job Scheduling Layer

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified [15, 21, 25].

Sub-job scheduling comes from the complex transfer from the job scheduling layer. It mainly includes job analysis, job generation and monitoring. Sub-job scheduling network is defined as:

$$S\_Grid_i = \{P_i, F_i, D_i, C_i, I_i, O_i, K_i, M_i\}$$

And

$i = 4$

which means

$$P4 = \{P_{start}^4, P_{end}^4, p_i^4 \mid 1 \leq i \leq 10\}$$

$$T4 = \{t_i^4 \mid 1 \leq i \leq 15\}$$

$$D_4 = \{C(p^4_{start}) \cup C(p^4_{end}) \cup d^4\}$$

Table 2 illustrates the indices used in this layer.

TABLE II INDICES EXPLANATIONS

d^4_i	Explanations
d^4_1	Jobs are initiated. $C(p^4_{start}) = \{d_4\}$, $C(p^4_{end}) = \{d_5, d_6, d_7\}$, $C(t^4_1) = \{d_4\}$, $C(t^4_3) = \{d^4_1\}$
d^4_2	Jobs are not initiated. $C(p^4_1) = \{d^4_1, d^4_2\}$, $C(t^4_2) = \{d^4_2\}$
d^4_3	Sub-jobs $C(p^4_2) = \{d^4_3\}$, $C(t^4_4) = \{d^4_3\}$
d^4_4	Sub-jobs are under running. $C(p^4_3) = \{d^4_4\}$, $C(t^4_5) = \{d^4_4, d^4_{12}\}$
d^4_5	Sub-jobs are waiting for check. $C(p^4_4) = \{d^4_5\}$, $C(t^4_6) = \{d^4_5\}$
d^4_6	Sub-jobs are correct. $C(p^4_5) = \{d^4_6, d^4_7, d^4_8\}$, $C(t^4_8) = \{d^4_6\}$
d^4_7	Sub-jobs are completed. $C(p^4_5) = \{d^4_6, d^4_7, d^4_8\}$, $C(t^4_9) = \{d^4_7\}$
d^4_8	Sub-jobs are failed. $C(p^4_5) = \{d^4_6, d^4_7, d^4_8\}$, $C(t^4_7) = \{d^4_8\}$
d^4_9	Sub-jobs are labelled. $C(p^4_6) = \{d^4_9\}$
d^4_{10}	Sub-jobs are checked. $C(p^4_7) = \{d^4_{10}, d^4_{11}\}$, $C(t^4_{13}) = \{d^4_{10}\}$
d^4_{11}	Sub-jobs are not checked. $C(p^4_7) = \{d^4_{10}, d^4_{11}\}$, $C(t^4_{11}) = \{d^4_{11}\}$
d^4_{12}	Sub-jobs TOKEN, $C(p^4_8) = \{d^4_{12}\}$, $C(t^4_5) = \{d^4_4, d^4_{12}\}$
d^4_{13}	Sub-jobs clean TOKEN. $C(p^4_9) = \{d^4_{13}\}$, $C(t^4_{12}) = \{d^4_4, d^4_{13}\}$
d^4_{14}	Sub-jobs have been done. $C(p^4_{10}) = \{d^4_{14}, d^4_{15}\}$, $C(t^4_{14}) = \{d^4_{14}\}$
d^4_{15}	Sub-jobs are under running with tasks. $C(p^4_{10}) = \{d^4_{14}, d^4_{15}\}$, $C(t^4_{15}) = \{d^4_{15}\}$

C. Task Scheduling Layer

Task scheduling layer is the sub layer of the sub-job scheduling. It is extended by the complex status transferring t^j_i , which contains the analysis of tasks and their monitoring. Let

$$i = 6, j = 4$$

$$P_{4,6} = \{p^{4,6}_{start}, p^{4,6}_{end}, p^{4,6}_i \mid 1 \le i \le 10\}$$

$$T_{4,6} = \{t^{4,6}_i \mid 1 \le i \le 15\}$$

$$D_{4,6} = \{C(p^{4,6}_{start}) \cup C(p^{4,6}_{end}) \cup d^{4,6}\}$$

$$d^{4,6} = \{d^{4,6}_i \mid 1 \le i \le 15\}$$

The status transferring logics and their relationships are illustrated in Fig 2.

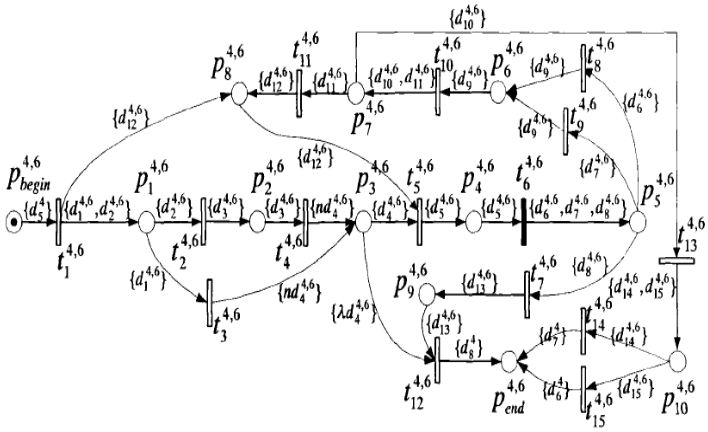


Fig. 2. Status Transferring and Relationships in Tasks Scheduling Layer

D. Sub-task Scheduling Layer

Sub-task scheduling layer is extended from the task scheduling layer according to t^j_i . Its mainly purpose is to analysis the tasks, arrange and generate sub-tasks as well as monitor their statuses and re-allocation [24].

Set

$$i = 6, j = 5$$

$$P_{4,6,6} = \{p^{4,6,6}_{start}, p^{4,6,6}_{end}, p^{4,6,6}_i \mid 1 \le i \le 14\}$$

$$T_{4,4,6} = \{t^{4,4,6}_i \mid 1 \le i \le 21\}$$

$$D_{4,4,6} = \{C(p^{4,4,6}_{start}) \cup C(p^{4,4,6}_{end}) \cup d^{4,4,6}\}$$

$$d^{4,4,6} = \{d^{4,4,6}_i \mid 1 \le i \le 15\}.$$

Table 3 reports the explanation of different parameters.

TABLE III EXPLANATIONS OF PARAMETERS IN SUB-TASK SCHEDULING LAYER

$d_i^{4,6,6}$	Explanations
$d_1^{4,6,6}$	Tasks are not initiated. $C(p_1^{4,6,6})=\{d_1^{4,6,6},d_2^{4,6,6}\}$, $C(t_3^{4,6,6})=\{d_1^{4,6,6}\}$
$d_2^{4,6,6}$	Tasks have been initiated. $C(p_1^{4,6,6})=\{d_1^{4,6,6},d_2^{4,6,6}\}$, $C(t_2^{4,6,6})=\{d_2^{4,6,6}\}$
$d_3^{4,6,6}$	Sub-tasks are under running. $C(p_2^{4,6,6})=\{d_3^{4,6,6}\}$, $C(t_4^{4,6,6})=\{d_3^{4,6,6}\}$
$d_4^{4,6,6}$	Query result of resources. $C(p_3^{4,6,6})=\{d_4^{4,6,6}\}$, $C(t_5^{4,6,6})=\{d_4^{4,6,6},d_{12}^{4,6,6}\}$
$d_5^{4,6,6}$	Select results of resources. $C(p_4^{4,6,6})=\{d_5^{4,6,6}\}$, $C(t_6^{4,6,6})=\{d_5^{4,6,6}\}$
$d_6^{4,6,6}$	Null resources. $C(p_5^{4,6,6})=\{d_6^{4,6,6},d_7^{4,6,6},d_8^{4,6,6}\}$, $C(t_8^{4,6,6})=\{d_6^{4,6,6}\}$
$d_7^{4,6,6}$	Sub-task layer. $C(p_5^{4,6,6})=\{d_6^{4,6,6},d_7^{4,6,6},d_8^{4,6,6}\}$, $C(t_9^{4,6,6})=\{d_7^{4,6,6}\}$
$d_8^{4,6,6}$	Selection label of sub-task. $C(p_5^{4,6,6})=\{d_6^{4,6,6},d_7^{4,6,6},d_8^{4,6,6}\}$, $C(t_7^{4,6,6})=\{d_8^{4,6,6}\}$
$d_9^{4,6,6}$	Sub-tasks are for scheduling. $C(p_6^{4,6,6})=\{d_9^{4,6,6}\}$
$d_{10}^{4,6,6}$	Sub-tasks are under running. $C(p_7^{4,6,6})=\{d_{10}^{4,6,6},d_{11}^{4,6,6}\}$, $C(t_{13}^{4,6,6})=\{d_{10}^{4,6,6}\}$
$d_{11}^{4,6,6}$	Sub-tasks are completed. $C(p_7^{4,6,6})=\{d_{10}^{4,6,6},d_{11}^{4,6,6}\}$, $C(t_{11}^{4,6,6})=\{d_{11}^{4,6,6}\}$
$d_{12}^{4,6,6}$	Sub-tasks are uncommon. $C(p_8^{4,6,6})=\{d_{12}^{4,6,6}\}$, $C(t_5^{4,6,6})=\{d_4^{4,6,6},d_{12}^{4,6,6}\}$
$d_{13}^{4,6,6}$	Sub-tasks are not labeled. $C(p_9^{4,6,6})=\{d_{13}^{4,6,6}\}$, $C(t_{12}^{4,6,6})=\{d_4^{4,6,6},d_{13}^{4,6,6}\}$
$d_{14}^{4,6,6}$	Sub-tasks have not scheduled yet. $C(p_{10}^{4,6,6})=\{d_{14}^{4,6,6},d_{15}^{4,6,6}\}$, $C(t_{14}^{4,6,6})=\{d_{14}^{4,6,6}\}$
$d_{15}^{4,6,6}$	Sub-tasks have scheduled. $C(p_{10}^{4,6,6})=\{d_{14}^{4,6,6},d_{15}^{4,6,6}\}$, $C(t_{15}^{4,6,6})=\{d_{15}^{4,6,6}\}$
$d_{16}^{4,6,6}$	All sub-tasks are completed. $C(p_{14}^{4,6,6})=\{d_{16}^{4,6,6},d_{17}^{4,6,6}\}$, $C(t_{20}^{4,6,6})=\{d_{16}^{4,6,6}\}$
$d_{17}^{4,6,6}$	Tasks with some sub-tasks uncompleted. $C(p_{14}^{4,6,6})=\{d_{16}^{4,6,6},d_{17}^{4,6,6}\}$, $C(t_{19}^{4,6,6})=\{d_{17}^{4,6,6}\}$
$d_{18}^{4,6,6}$	Sub-tasks are reallocated successfully. $C(p_{11}^{4,6,6})=\{d_{18}^{4,6,6},d_{19}^{4,6,6}\}$, $C(t_{14}^{4,6,6})=\{d_{18}^{4,6,6}\}$
$d_{19}^{4,6,6}$	Sub-tasks are reallocated unsuccessfully. $C(p_{11}^{4,6,6})=\{d_{18}^{4,6,6},d_{19}^{4,6,6}\}$, $C(t_{16}^{4,6,6})=\{d_{19}^{4,6,6}\}$
$d_{20}^{4,6,6}$	Sub-tasks run after reallocation. $C(p_{13}^{4,6,6})=\{d_{20}^{4,6,6}\}$, $C(t_{15}^{4,6,6})=\{d_{20}^{4,6,6}\}$
$d_{21}^{4,6,6}$	Sub-tasks clean label. $C(p_{12}^{4,6,6})=\{d_{21}^{4,6,6}\}$, $C(t_{17}^{4,6,6})=\{d_{21}^{4,6,6}\}$

The four-layer CHPN scheduling algorithm describes specific implementation processes that the user submits a job. It is based on workflow-based approach to relate to the steps and the specific properties of interrelated management, which aims to decompose the scheduling complexity into different levels. Therefore, different levels of granularity are managed by the corresponding layer and the process is based on variable structure, in which a task level can be extended [16]. The sub-task statuses transfer is shown in the following Fig. 3.

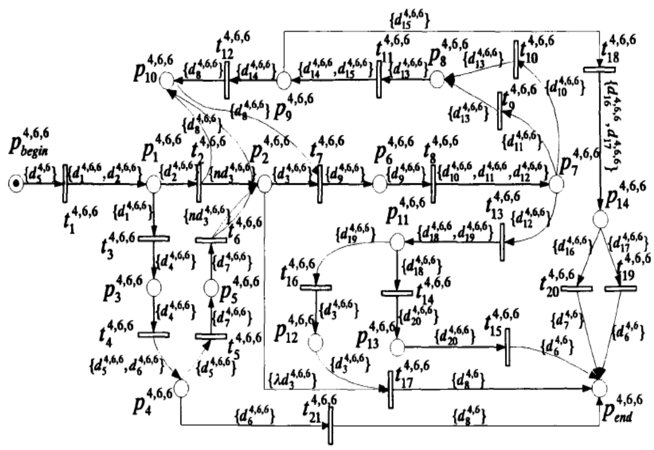


Fig.3. Status Transfer in Sub-task Scheduling Layer

IV. SIMULATION AND ANALYSIS

The experiments are carried out through a DMGS network which is based on the reachable tree model. The initiatives of this model is

$$M_0=\{1,0,0,0,0,0,0,0,0,0,0,0\}$$

while the finish status is

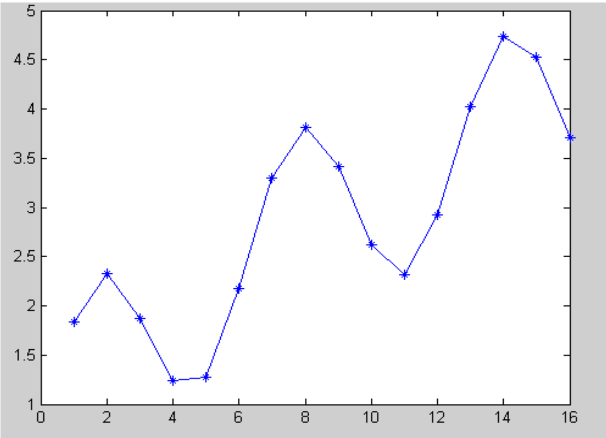
$$M_e=\{0,0,0,0,0,0,0,0,0,0,0,1\}.$$

The number of jobs is 16 in the simulation. The sub-jobs is 15 with 16 input tasks. Each task contains 21sub-tasks. The objective function is defined as:

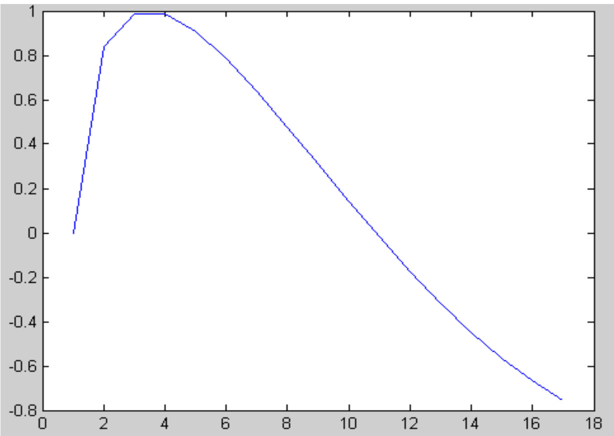
$$\sum_{r=1}^{16} Q_r(t^r)$$

which subjects to

$$\max(\tau_r(t^1),\ldots\tau_r(t^{16})).$$



(a) Status and Time Relation



(b) Algorithm Convergnece

Fig. 4. Experiment Results

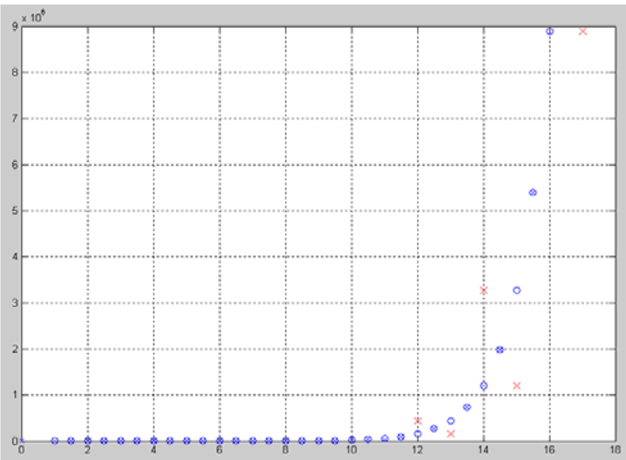


Fig.5. Deviation Analysis

The experiment environment is based on a computer with processor: Intel (R) Core (TM)2 CPU 6600@2.40GHz and RAM: 4.00GB (3.49 GB usable). System type is 32-bit Operating System of Windows 7 Enterprise. The configuration of the experiment is under Matlab 2008 through the Petri Net Toolbox for Matlab which is for simulation, analysis and design of discrete event systems, based on the Petri Net model proposed in this paper. All the initiations are carried out in this toolbox so that the experiments results could be achieved and detailedly discussed as follows.

The reachable tree model has a critical path that can generate the optimal results [2]. Matlab simulation is used in this experiment to carry out the CHPN scheduling. Results are achieved as the following figures.

Fig. 4 (a) illustrates the comparison of status transferring and the cost of time. The time cost of the algorithm proposed in this paper is [1.8S，4.75S]. (b) shows the convergence of the algorithm. It is obvious that the algorithm solution space is bounded with the problem size increases, further enhance convergence will be achieved.

The CHPN scheduling algorithm features large number of calculation or transfer, long-time running performance [17]. After the top to down hierarchy operations, parallel processes are achieved through reachable principles which means from the very beginning status, there are some statuses can be achieved during the scheduling processes [18]. The experiment uses 16 statuses with the TOKEN subjects to

$$\sum_{i=1}^{16} C(P_i^J) = 1$$

Fig. 5 shows the deviation which marked with ×, which indicates the deference of O.

The deviation is

$$\sum_{i=1}^{n} |e_i| \ = 3.4$$

While the deviation ratio is 3.125%. Table 4 shows the time cost and deviation achieved from the experiments.

TABLE IV TIME COST AND DEVIATION

Layer Amount	Time Cost (s)	Transfer Status	Deviation Ratio
0	170	32	5.14%
1	129	43	5.01%
2	94	79	3.78%
4	77	106	3.125%

V. CONCLUSION

Data mining grid scheduling (DMGS) is a dynamic scheduling with characteristics that job size is influenced by resources. The paper defines DMGS into a single workflow model, using colored-hierarchy Petri Nets (CHPN) variable structure algorithm. The algorithm is divided into four layers: job scheduling, sub-job scheduling, task scheduling and the sub-task scheduling. From top to bottom with each level is based on CHPN, the scheduling complexity is alleviated.

Experimental results show that the algorithm can effectively solve the DMGS problem with acceptable effectiveness and efficiency.

In this paper, CHPN based on the actual production process is used to solve the DMGS problems. There are some limitations which should be improved in the future work. First of all, this paper does not take into account uncertainties such as the task deadlock, the process of disturbances etc. Therefore, this paper can be extended in adding the now assumptions. Secondly, all the scheduling processes are carried out in DMGS, which does not consider the state changes in the production processes. Finally, the actual real-time information feedback relationship is not considered as well. Thus, the qualitative and quantitative analysis will be carried out in the future work.

ACKNOWLEDGMENT

The author would like to thanks the finance support from different parties. Technical supports from our research team will be greatly appreciated in terms of data analysis, model construction and experiment executions.

REFERENCES

[1] Cannataro, M., D. Talia, and P. Trunfio, Distributed data mining on the grid. Future Generation Computer Systems, 2002. 18(8): p. 1101-1112.

[2] Stankovski, V., et al., Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. Future Generation Computer Systems, 2008. 24(4): p. 259-279.

[3] Han, J. and M. Kamber, Data mining: concepts and techniques2006: Morgan Kaufmann.

[4] Chen, C.S., C.H. Lin, and H.Y. Tsai, A rule-based expert system with colored Petri net models for distribution system service restoration. IEEE Transactions on Power Systems,, 2002. 17(4): p. 1073-1080.

[5] Wu, N., et al., Petri Net-Based Scheduling of Single-Arm Cluster Tools With Reentrant Atomic Layer Deposition Processes. IEEE Transactions on Automation Science and Engineering,, 2011. 8(1): p. 42-55.

[6] Chen, J.H., et al., Petri-net and GA-based approach to modeling, scheduling, and performance evaluation for wafer fabrication. IEEE Transactions on Robotics and Automation,, 2001. 17(5): p. 619-636.

[7] Huang, C.L., M.C. Chen, and C.J. Wang, Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 2007. 33(4): p. 847-856.

[8] Zhou, M.C., Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach. IEEE Transactions on Semiconductor Manufacturing,, 1998. 11(3): p. 333-357.

[9] Wang, W., J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. 1997. Citeseer.

[10] Cannataro, M., D. Talia, and P. Trunfio, Distributed data mining on the grid. Future Generation Computer Systems, 2002. 18(8): p. 1101-1112.

[11] Stankovski, V., et al., Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. Future Generation Computer Systems, 2008. 24(4): p. 259-279.

[12] Han, J. and M. Kamber, Data mining: concepts and techniques2006: Morgan Kaufmann.

[13] Chen, C.S., C.H. Lin, and H.Y. Tsai, A rule-based expert system with colored Petri net models for distribution system service restoration. IEEE Transactions on Power Systems,, 2002. 17(4): p. 1073-1080.

[14] Chaouiya, C., E. Remy, and D. Thieffry, Petri net modelling of biological regulatory networks. Journal of Discrete Algorithms, 2008. 6(2): p. 165-177.

[15] Pastor, E., et al., Petri net analysis using boolean manipulation. Application and Theory of Petri Nets 1994, 1994: p. 416-435.

[16] Best, E. and R. Devillers, Sequential and concurrent behaviour in Petri net theory. Theoretical Computer Science, 1987. 55(1): p. 87-136.

[17] Peterson, J.L., Petri Net Theory and the Modeling of Systems. PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, NJ 07632, 1981, 290, 1981.

[18] Reisig, W., Petri nets. Modeling in Systems Biology, 2011: p. 37-56.

[19] Wu, N., et al., Petri Net-Based Scheduling of Single-Arm Cluster Tools with Reentrant Atomic Layer Deposition Processes. IEEE Transactions on Automation Science and Engineering,, 2011. 8(1): p. 42-55.

[20] Hamadi, R. and B. Benatallah. A Petri net-based model for web service composition. 2003. Australian Computer Society, Inc.

[21] Chen, J.H., et al., Petri-net and GA-based approach to modeling, scheduling, and performance evaluation for wafer fabrication. IEEE Transactions on Robotics and Automation,, 2001. 17(5): p. 619-636.

[22] Lohmann, N., A feature-complete Petri net semantics for WS-BPEL 2.0. Web Services and Formal Methods, 2008: p. 77-91.

[23] Li, Z.W., M.C. Zhou, and N.Q. Wu, A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2008. 38(2): p. 173-188.

[24] Ghezzi, C., et al., A unified high-level Petri net formalism for time-critical systems. IEEE Transactions on Software Engineering, 1991: p. 160-172.

[25] Huang, C.L., M.C. Chen, and C.J. Wang, Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 2007. 33(4): p. 847-856.

[26] Zhou, M.C., Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach. IEEE Transactions on Semiconductor Manufacturing,, 1998. 11(3): p. 333-357.

[27] Wang, W., J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. 1997. Citeseer.